Quality Guaranteed?
A discussion on standardization and certification of information systems development

G.J. van der Pijl, Tilburg University
J.G. Verrijdt, Rabobank Nederland
G.J.P. Swinkels, Rabobank Nederland, University of Amsterdam

POB 90153
5000 LE Tilburg
The Netherlands
tel. 31-13-662138
E-mail: PIJL@KUB.NL

1. Introduction

If a company wants to maintain its strategic position it can be necessary to improve the quality of its business processes and to communicate these improvements to the customers. In European countries the ISO 9000 standards and accompanying certificates are becoming a more and more popular way to do this. The ISO 9000 standards [ISO87] are also used by software houses and software development departments within companies that try to get their quality systems certified. More specifically for systems development the ISO 9000-3 [ISO91] standard has been specified. A growing number of information systems developers use standards as a way of implementing their quality systems.

There is, however, a lot of discussion on these standards and the certificates accompanying them. Many companies seem to be eager to get the certificate because of market pressure. This often means that a minimum set of procedures and quality handbooks is set up without a quality system really being implemented, or quality awareness being created among the employees. In those cases the possession of the certificates does not guarantee the quality of the software production process or a reasonable price/performance ratio for the products delivered to the client. Even if a company maintains a quality system according to the standards of ISO 9000, the quality of the final product can never be guaranteed. And worse still: adhering to strict software-quality standards can in some circumstances be counterproductive because in many cases the software development process has to be attuned to the specific type of system that has to be produced and the specific systems development environment in which the production takes place.

In this paper we discuss the use of standards in the control of the quality of the information systems development process. The ISO model, which is the most popular standard in Europe and Japan, is compared with the Capability Maturity Model (CMM) [Hump88], [Hump89], [Paul91], [Paul93] which is becoming the main de facto standard in the USA.

When looking at standards like these, we have to realize that they serve a dual purpose. On the one hand they can be used to provide guidance to organizations installing quality systems that help them to produce goods and services of the desired quality. On the other hand they are the basis for certification of quality systems. Quality certificates are meant to give buyers of goods and services an impression of the quality of the suppliers and thus indirectly of the quality of the products and services supplied [Rugg93]. Problems with standards can arise in either these fields.

We show that CMM solves some of the problems of the ISO model. But CMM also has shortcomings. Therefore we indicate the direction in which software quality assurance and software quality standards have to develop in order to improve the quality of the information systems delivered by the development process.

2. The ISO 9000 standards

The set of international standards ISO 9000-9004 describes how companies can implement quality systems in their organizations. Once a quality system is in place the organization can have it certified by an independent institution. The certificates can then be used to show the customers that the company adheres to externally verified quality standards.

The philosophy of the standards is consistent with the developments in the theory on quality management [Jura88]. The focus is shifting from controlling the quality of the final product to controlling the quality of the production processes from which the product arises (see [Pijl94]). The essence of the ISO 9000 standard is that a company should visibly control all aspects of the business in order to guarantee a minimum level of quality of its products. Visibility is realized by describing the quality system in a Quality Handbook. The standard does not describe what is the most effective and efficient way of controlling the business processes. It just demands that the organization maintains sets of quality procedures and guidelines without exactly specifying them. This is both the strength and the weakness of the standard. The open character makes it applicable to many different production processes. But for producers of goods and services it makes it hard to decide on the specific type of measures to be taken in specific situations. And for purchasers it means that a quality certificate does not necessarily indicate that the suppliers' quality system is well attuned to the specific product and market situation. This is the main reason for the development of more and more specific versions and interpretations of the standard for different types of industries.

For systems development we have the ISO 9000-3 standard: 'Guidelines for the development, supply and maintenance of software'. In this standard the life-cycle of information systems is taken as the process to be controlled. However, it remains a very general description of the procedures and guidelines for quality management.

ISO 9000 is very popular both in the United Kingdom and in the Netherlands and the number of applications in Germany is rising. There is, however, a lot of criticism. This criticism stems partly from the fact that companies try to get ISO 9000 certificates for reasons of marketing rather than for reasons of (total) quality management. But even if we accept that companies use standards to really improve their production processes, it is possible to be critical of the contents and

construction of the standards. Important criticisms concerning the use of ISO 9000-3 for systems development are:

- The general nature of the standard makes it necessary to translate it into specific quality measures in each organization. The standard only demands that quality procedures are installed. It does not, however, indicate what the *correct* procedures are. Thus the standard cannot be used as a detailed blueprint for the construction of quality systems in practical situations. It is also hard to use the standard as a norm when judging the appropriateness of quality systems in specific situations. This implies that possession of the ISO 9000 certificate is only weak proof of the quality of a software supplier. In many cases additional audits will be necessary to properly assess this quality;
- The standard is specified on the level of quality systems for systems development organizations and gives only limited attention to quality measures on the level of specific development projects;
- ISO 9000 recognizes only one certifiable level of quality while in practice different quality levels may be advisable for different situations. It is quite possible that, within a company with several systems development departments, not all of these have to operate at the same quality level because they are developing different types of products. Levels of quality would also be useful to indicate the stages along a growth path that a systems development organization could follow in improving the quality of the work. It is also possible that buyers of information systems do not look for the same quality level for each and every information system they buy;
- The emphasis on strict procedures and the documentation of these in manuals easily leads to a bureaucratic type of behaviour instead of a growing awareness of the importance of quality for the organization. Attention for new insights and methodologies in systems development may be hampered. A good example is Boehm’s "spiral model" [Boeh88] that tries to cope with uncertainty in systems development but thus clashes with the rules of ISO 9000-3;
- In some cases throughput-time may be of more importance for the development of information systems than superb quality. Quality standards, however, leave very little room for the principles of rapid application development.

3. The Capability Maturity Model

The CMM as put forward by the Software Engineering Institute builds on the ’maturity model’ presented by Humphrey [Hump89].

The maturity model presents a growth theory according to which the quality level of a systems development organization can grow along a given growth path. The gist of the model is that several quality levels for the systems development process can be recognized. An organization can go from one stage to another and thus grow from a situation with no quality management to a mature situation with a very high level of quality control, as described in exhibit 1.

level CMM	description
1. "initial" processes	- ad hoc, sometimes chaotic processes - project succes not guaranteed
3. "defined" processes	- process documented - process standardized - tailored standards used for each project
4. "managed" processes	- process understood - process measured - process controlled
5. "optimizing" processes	- focus on process improvement - focus on rapid technology updating

exhibit 1: The quality levels of CMM

Each level is determined by a set of key process areas which in turn consist of sets of key practices [Webe91]. By measuring which key practices are implemented the maturity level of a systems development organization can be measured. A structure for such a measurement is given in exhibit 2. As shown there, each key process can be judged as being sufficiently (S), partly (P) or insufficiently (I) implemented. The total set of key practices is specifically aimed at the systems development organization and is much more detailed than the prescriptions of ISO-9000, thus leaving much less room for interpretation.

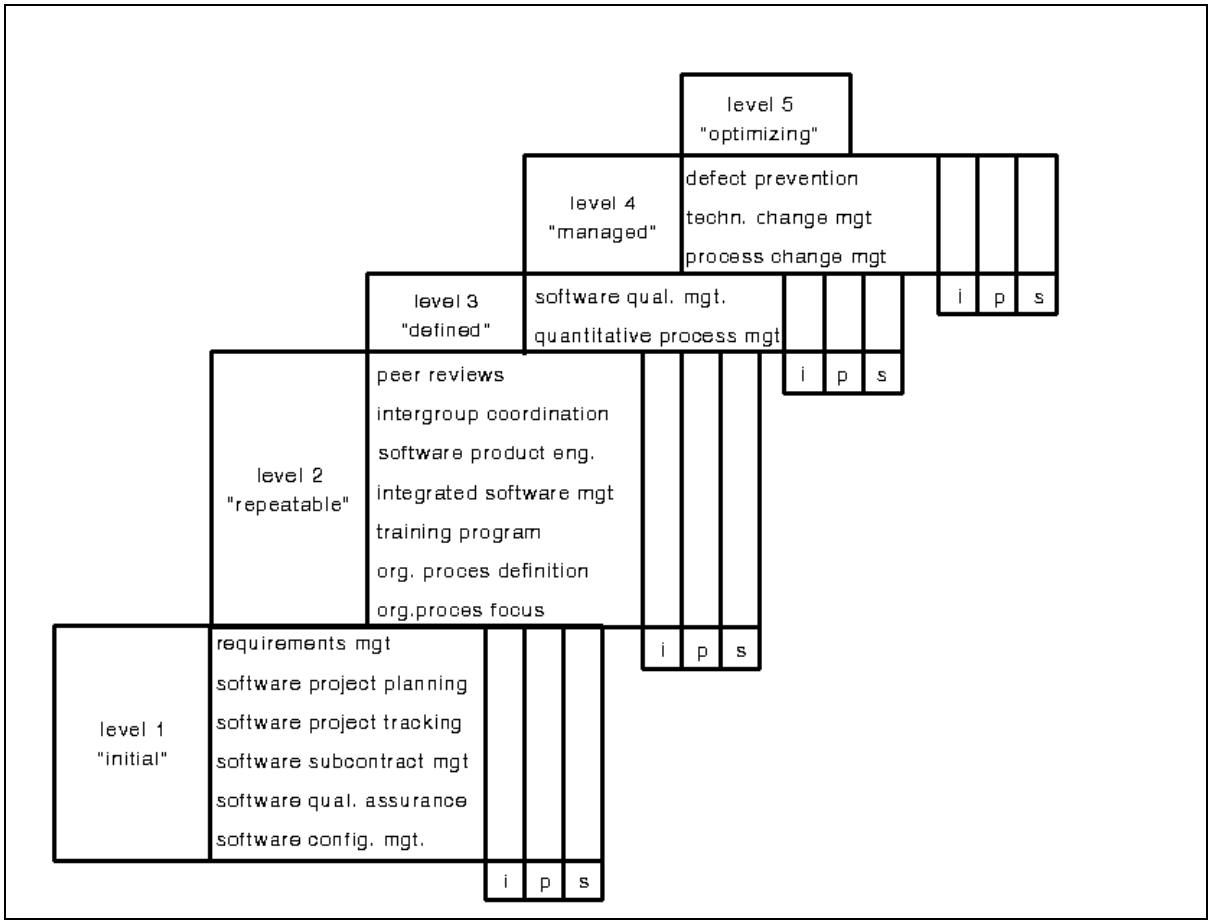

exhibit 2: assessing the quality level of an organization

Each quality level can thus be seen as a well-defined stage on the way to a mature organization. From each stage a set of well-defined small steps leads up to the next higher quality level. Of course the leading thought behind the model is that a higher degree of maturity leads to better results in terms of defects, costs, throughput times and quality of the systems delivered. Recent research [Dreh93] shows that the CMM's quality levels cannot exactly be found in the real world. Possibly a more gradual pattern of growth would be a more realistic hypothesis. This does not, however, fundamentally change the idea of CMM as an elaborate example of how to install quality management in systems development in a series of consecutive steps. It might, however, influence the idea that quality of a software suppliers development process can be assessed by looking at a detailed set of key processes and key practices and assessing at which stage of the CMM model the supplier is. There may be other possible growth paths that also enable suppliers to deliver software of the desired quality. The quality of these suppliers would not be recognized if their business processes are only compared to the levels of CMM.

4. Comparing ISO 9000-3 and CMM

Both ISO 9000 and CMM are widely accepted standards. CMM is a de facto standard in the USA. In Europe and Japan the number of companies using CMM is growing. ISO 9000-3 is recognized all over the world. Both standards can be considered stable in the sense that they have widely been used for a number of years and "teething troubles" have been cured. There are possibilities for computer-supported use of both standards but they are not very well developed. Our research [Verr93], in which the demands of ISO 9000-3 are compared with CMM's key practices, shows that there is much overlap between the two approaches. This becomes clearest at the second and third levels of CMM. Almost all articles of the ISO standard can be translated into key practices at these two levels. Implementing CMM up to the third level and completing it with only a few additional items (5.2.2 "contract items on quality"; 5.8 "acceptance"; 5.9 "replication, delivery and installation"; 6.2.4 "document changes"; 6.8 "included software") from ISO 900-3 make certification according to ISO 9000 possible.

However, there are also important differences between the approaches:

- CMM is specially aimed at software development. It models the development process in a much more detailed way than ISO 9000. The quality characteristics of the levels in CMM are described in much more detail than the quality characteristics of ISO 9000-3. Therefor it is much easier to use CMM as a guideline for building quality systems in software development.
- At the higher levels of CMM differences between software projects are explicitly taken into account. Accumulation of statistical data on success and failure factors on the project level makes it possible to attune the quality system to the project at hand.
- The ISO standards prescribe one fixed level of quality management. CMM points out that different levels are possible but does not prescribe which level a systems developer has to choose. The idea is that this is determined by the specific situation in which the systems development takes place. The levels 4 and 5 of the CMM model clearly stand for higher quality than does ISO 9000-3.
- The structuring of the quality characteristics in CMM makes it easier to discuss quality matters.

From the description above it will be clear that CMM solves some of the problems associated with ISO 9000. The more detailed specification of quality characteristics makes it much easier to use CMM as a blueprint for quality systems in systems development organizations. Because of the explicit attention to the relevant aspects of systems development it is also much easier attune it to different circumstances. Quality assessment on the basis of CMM, therefore, gives the purchaser much more certainty that the quality system is well attuned to the demands of the situation. The fact that CMM recognizes "maturity levels" for quality systems makes it possible for software development organizations to improve their quality programs progressing from one maturity level to the next. Organizations that have reached the third maturity level of CMM (the "defined" stage) comply in most aspects with the ISO 9000-3 standard (see exhibit 3). But CMM permits one to stop at a lower level or to aim at higher levels of process quality. Thus it can be more easily attuned to different circumstances. On the other hand purchasers can select systems development organizations with a quality level fitting their purposes.

Since it is not completely clear that the growth path to quality that is indicated by CMM is the only possible growth path for reaching quality, one should be careful to use CMM as the exclusive basis for certification. It might very well be that organizations following a different growth path also deliver good quality software [Boll91].

5. Expected trends

In our view we should focus on the quality of the development process itself and not only on the quality system. The quality system is an integral part of this process. The measures that ensure the quality of an information system that has to be developed, have to be tuned to the characteristics of that system and to the circumstances under which the development takes place. Software projects have to be analyzed in relation to the strength and weaknesses of both the development organization and the user-organization [Giel92]. This analysis determines the specific set of quality measures to be taken. But if the set of measures to be used can differ from project to project this makes it very difficult to use a standard quality assurance system. We will have to find a balance between specifying a set of minimum requirements valid for all projects and leaving enough room for the particularities of individual projects. In finding this balance we will have to take account of a number of aspects that we will discuss briefly.

- A complete quality system takes into account all factors that explain the quality of the final product. A variety of aspects, such as the quality of the people involved, the methods and tools used and the commitment of the management, influence the result. Empirical evidence shows that we do not know all the factors and causal relationships that explain the final results [Cape86]. Only a limited number of factors is taken into account in the standards. In order to develop good quality software, however, we have to take all factors into account.
- Measures to improve quality are very divers (preventive, detective, corrective) and can partly replace one another. If, for example, not all members of a project team can be instructed in the use of a method, we will have to pay more attention to testing the final product and feeding back the results of the tests to the project members.
- Measures taken for a specific project can conflict with the normal standards. Using a standard method has the advantage that all the members of the organization are familiar with it. But using several methods makes it possible to take care of the specifics of the project involved. Differences in steps and deliverables make it difficult in that case to compare projects and thus gain experience.
- The more quality systems are implemented in organizations the more probable it is that there are projects in which several parties have there own quality systems. Taking into account all the requirements of these different systems will often be very complicated and costly.

- The present quality standards are mainly oriented towards standardizing activities and procedures. In practice, however, projects are also controlled by controlling deliverables and the capabilities of members of the project team. The specific mix of control mechanisms is determined by the circumstances.
- Project- and organization-specific quality systems can only be implemented and certified if products and processes can be measured with a clear and widely accepted set of metrics. Though work in this direction is going on in the ESPRIT-AMI project, there is no such generally accepted set of metrics.
- Information technology and methods of information systems development are changing quickly. International standards generally develop at a much slower pace. Thus it is a problem to continually include new insights and developments in the standards.

There are a number of trends that support the growing importance of quality assurance such as outsourcing and facilities management, fixed price, fixed date, but also fixed quality contracts between purchasers and developers of information systems, service level agreements etc. There are also some trends that support the development of more situation specific norms, such as norms for specific areas (e.g. detachment), development of quality metrics (e.g. using function points) and specifying software quality [ISO 9126].

For the time being most software developing-organizations can still improve their quality by using the existing standards wisely and adapting them to the needs of specific situations. This also means that buyers of software should realize that adhering to the existing standards (that is: what is certified by a certificate) is not always the best thing do. Well founded deviations from the standards should be accepted. Furthermore we have to keep in mind that standardization of processes is not the only way to guarantee quality. Other coordination mechanisms like standardization of skills, standardization of outputs and communication between all concerned must also be used [Pijl90].

The specification of a quality system is a matter of weighing costs and benefits, on the level of organizations as a whole as well as on the level of specific projects. Important questions are:

- Are our clients obliging us to comply with standards (such as, for example, AQAP-norms for the military sector or Lloyds for the steel industry)?

- Does a certified quality system result in a growing turnover because of its commercial importance?
- Does such a formalized approach fit in the culture of the organization?
- Does a well-balanced quality system on the project level make it possible to offer our clients better contracts with lower risks (e.g. fixed date, fixed price, fixed quality)?
- How certain is it that we can profit from our investments in quality systems in the future?
- Does reduction of the number of errors reduce costs (for example because of a smaller number of fixes or lower insurance premiums)?

Advantages and disadvantages of improvements in the systems development process have to be weighed against each other and against the costs of implementation [Dion93]. Conclusions can differ for different parts of the organization.

In the long run efforts have to be directed at formulating more specific standards for specific situations. An alternative could be to develop standards that contain substandards relating to different circumstances.

6. Summary and conclusions

At present there is a conflict between the desire for better quality systems resulting in measurable improvements of the systems development process and the way in which this is implemented by means of standards. Quality certificates can thus give a wrong impression of the real capabilities of an organization. This should not lead, however, to an abolition of standards, as is suggested by some authors. The majority of organizations can improve by making (sensible) use of the standards. But there is a need for improving the standards.

In the short term the quality of systems development can be improved by an approach that is specially constructed for assessing systems development environments, which supports measurable improvement of the development process and which supports the choice and implementation of actions. Although not perfect, CMM offers more possibilities in this respect than does ISO 9000-3.

In the longer term standards and certificates will have to take into account the diversity that exists in the real world. Standards and certificates will have to be attuned to this diversity. We think that this process will take some time and that therefore we will have to carefully assess the value of quality certificates in different situations.

references:

[Boeh88] Boehm, B.W., A spiral model of software development and enhancement, *Computer* 21 NO.5, P.61-72, 1988.

[Boll91] Bollinger, T. and McGowan, C., A critical look at software capability evaluations, *IEEE Software* July 1991, p. 25-41.

[Cape86] Capers Jones, *Programming productivity*, McGraw-Hill, 1986.

[Dreh93] Drehmer, D. en Daklava, S., Measuring software engineering maturity: a Rash calibration, *Proceedings of the fourteenth internationational conference on information systems*, Orlando, December 1993.

[Dion93] Dion, R., Process improvement and the corporate balance sheet. *IEEE software*, July 1993.

[Giel92] Gielen L.J.M.W., Swinkels G.J.P., Kwaliteitsbeheersing bij de ontwikkeling van informatiesystemen via knelpuntenanalyse (Analysing strengths and weaknesses as a means of quality assurance for the development of information systems), *Handboek A.I.V.*, November 1992.

[Hump88] Humphrey W.S., Characterizing the software process, *IEEE Software*, march 1988

[Hump89] Humphrey W.S., *Managing the software process*, Addison Wesley, 1989

[ISO87] *ISO 9000-9004 quality management and quality assurance standards*, 1987

[ISO91] ISO 9000-3, Quality management and quality assurance standards, part 3: *Guidelines for the application of ISO 9001 to the development, supply and maintenance of software*, 1991

[ISO91.1] *ISO 9126 Software Product Evaluation-Quality Characteristics and Guidelines for their Use*, International Organization for Standardization, Geneva, 1991

[Jura88] Juran, J.M., *Juran on planning fot quality*, McMillan, New York, 19988.

[Paul91] Paulk M.C., Curtis B., Chrissis M.B., *Capability maturity model for software*, Software engineering institute, CMU/SEI-91-TR-24, DTIC Number AD240603, august 1991

[Paul93] Paulk M.C., Curtis B., Chrissis M.B.,Weber C.V., *Capability maturity model for software*, Version 1.1. Software engineering institute, CMU/SEI-93-TR-24, august 1993

[Pijl90] Pijl G.J. van der, Weterings P.L.M., Standaardisatie in systeemontwikkeling (Standardization and systems development), *Maandblad voor bedrijfsadministratie*

en organisatie, December 1990.

[Pijl94] Measuring the strategic dimensions of the quality of information, *The Journal of Strategic Information Systems*, vol. 3 nr.3, 1994

[Rugg93] Rugg D., Using a capability evaluation to select a contractor, *IEEE Software*, July, 1993.

[Verr93] Verrijdt J.G., *Het beheersen van kwaliteitsverbetering: groeien van Murphy naar Humphrey* (Controlling quality improvements: developping from Murphy to Humphrey)

[Webe91] Weber C.V., Paulk M.C., Wise C.J., Withey J.V., *Key practices of the capability maturity model*, Software engineering institute, CMU/SEI-91-TR-25, August 1991.